

Script(Create DB)

```
Type "help" for help.
[
postgres=# drop database if exists credit_cards_sl;
create database credit_cards_sl;

\c credit_cards_sl;

--TYPES
create type FullName as (name varchar, firstSurname varchar, secondSurname varchar);
create type FiscalAddress as (street varchar, number int, city varchar, postalCode int, country varchar);
create type CardTypeName as enum ('Standard', 'Gold', 'Premium');
create type CardLimit as enum ('1000', '2000', '3000');
create type CardType as (creditLimit CardLimit, paymentTerms int, type CardTypeName);

--DOMAINS
create domain Age int check(value > 0 and value < 125);

--TABLES
create table Client (id serial primary key, billing int, registerDate date);
create table CreditCard (id serial primary key, associatedAccount int unique not null, expiryDate date not null, cardType CardType not null default ('1000', 4, 'Standard'), client_id int not null) with oids;
create table Person (nif varchar unique, fullName FullName, age Age) inherits (Client) with oids;
create table Company (cif varchar unique, fiscalAddress FiscalAddress) inherits (Client) with oids;
postgres=# create database credit_cards_sl;
CREATE DATABASE
postgres=#
postgres=# \c credit_cards_sl;
You are now connected to database "credit_cards_sl" as user "andrey.melnikov".
credit_cards_sl=#
credit_cards_sl=# --TYPES
credit_cards_sl=# create type FullName as (name varchar, firstSurname varchar, secondSurname varchar);
CREATE TYPE
credit_cards_sl=# create type FiscalAddress as (street varchar, number int, city varchar, postalCode int, country varchar);
CREATE TYPE
credit_cards_sl=# create type CardTypeName as enum ('Standard', 'Gold', 'Premium');
CREATE TYPE
credit_cards_sl=# create type CardLimit as enum ('1000', '2000', '3000');
CREATE TYPE
credit_cards_sl=# create type CardType as (creditLimit CardLimit, paymentTerms int, type CardTypeName);
CREATE TYPE
credit_cards_sl=#
credit_cards_sl=# --DOMAINS
credit_cards_sl=# create domain Age int check(value > 0 and value < 125);
CREATE DOMAIN
credit_cards_sl=#
credit_cards_sl=# --TABLES
credit_cards_sl=# create table Client (id serial primary key, billing int, registerDate date);
CREATE TABLE
credit_cards_sl=# create table CreditCard (id serial primary key, associatedAccount int unique not null, expiryDate date not null, cardType CardType not null default ('1000', 4, 'Standard'), client_id int not null) with oids;
CREATE TABLE
credit_cards_sl=# create table Person (nif varchar unique, fullName FullName, age Age) inherits (Client) with oids;
CREATE TABLE
credit_cards_sl=# create table Company (cif varchar unique, fiscalAddress FiscalAddress) inherits (Client) with oids;
CREATE TABLE
credit_cards_sl=# █
```

Insert

```
credit_cards_sl=# create or replace function pseudo_fk_for_creditcard() returns trigger as $$
credit_cards_sl=# begin
credit_cards_sl=#   if not exists(select id from client where id = new.client_id) then
credit_cards_sl=#       raise exception 'No such person: %', new.client_id;
credit_cards_sl=#   end if;
credit_cards_sl=#   return new;
credit_cards_sl=# end;
credit_cards_sl=# $$ language plpgsql;
CREATE FUNCTION
credit_cards_sl=#
credit_cards_sl=# --TRIGGERS
credit_cards_sl=# create trigger pseudo_fk_for_creditcard_trigger before insert or update on creditcard
credit_cards_sl=# for each row execute procedure pseudo_fk_for_creditcard();
ERROR:  trigger "pseudo_fk_for_creditcard_trigger" for relation "creditcard" already exists
credit_cards_sl=#
credit_cards_sl=# --INSERTIONS
credit_cards_sl=# --PERSONS
credit_cards_sl=# insert into Person values (default, 1200, now(), 'X1238754Q', ('Raul', 'Ortega', 'Barcells'), 23);
INSERT 16684 1
credit_cards_sl=# insert into Person values (default, 1150, now(), 'X1233254Q', ('Laura', 'Silva', 'Rubio'), 25);
ERROR:  duplicate key value violates unique constraint "person_nif_key"
DETAIL:  Key (nif)=(X1233254Q) already exists.
credit_cards_sl=# insert into Person values (default, 950, now(), 'X1234554Q', ('Silvia', 'Tigret', 'Gimenez'), 19);
ERROR:  duplicate key value violates unique constraint "person_nif_key"
DETAIL:  Key (nif)=(X1234554Q) already exists.
credit_cards_sl=# --COMPANIES
credit_cards_sl=# insert into Company values (default, 1500, now(), 'S0794867B', ('Lluís Figueras', 544, 'Barcelona', 08017, 'Espanistan'));
ERROR:  duplicate key value violates unique constraint "company_cif_key"
DETAIL:  Key (cif)=(S0794867B) already exists.
credit_cards_sl=# insert into Company values (default, 125000, now(), 'S0794677B', ('Fleming', 344, 'Barcelona', 08014, 'Espanistan'));
ERROR:  duplicate key value violates unique constraint "company_cif_key"
DETAIL:  Key (cif)=(S0794677B) already exists.
credit_cards_sl=# insert into Company values (default, 50000, now(), 'S0754677B', ('Gran Via', 542, 'Barcelona', 08013, 'Espanistan'));
ERROR:  duplicate key value violates unique constraint "company_cif_key"
DETAIL:  Key (cif)=(S0754677B) already exists.
credit_cards_sl=# --CREDIT CARDS
credit_cards_sl=# insert into CreditCard values (default, 51230, now() + interval '5 YEAR', default, 1);
ERROR:  No such person: 1
CONTEXT:  PL/pgSQL function pseudo_fk_for_creditcard() line 4 at RAISE
credit_cards_sl=# insert into CreditCard values (default, 51234, now() + interval '10 YEAR', ('2000', 12, 'Gold'), 2);
ERROR:  duplicate key value violates unique constraint "creditcard_associatedaccount_key"
DETAIL:  Key (associatedaccount)=(51234) already exists.
credit_cards_sl=# insert into CreditCard values (default, 51231, now() + interval '5 YEAR', ('3000', 12, 'Standard'), 3);
INSERT 16691 1
credit_cards_sl=# insert into CreditCard values (default, 51232, now() + interval '10 YEAR', ('3000', 2, 'Premium'), 4);
ERROR:  duplicate key value violates unique constraint "creditcard_associatedaccount_key"
DETAIL:  Key (associatedaccount)=(51232) already exists.
credit_cards_sl=# insert into CreditCard values (default, 51235, now() + interval '5 YEAR', ('2000', 15, 'Gold'), 5);
INSERT 16693 1
credit_cards_sl=# insert into CreditCard values (default, 51236, now() + interval '10 YEAR', ('3000', 3, 'Premium'), 6);
ERROR:  duplicate key value violates unique constraint "creditcard_associatedaccount_key"
DETAIL:  Key (associatedaccount)=(51236) already exists.
credit_cards_sl=# █
```

Query

```
credit_cards_sl=# select id, associatedAccount, expiryDate from CreditCard where expiryDate < now() + interval '5 DAY';
id | associatedaccount | expirydate
-----+-----+-----
(0 rows)

credit_cards_sl=# select nif, fullName from Person where billing > 1000;
nif | fullName
-----+-----
X1238754Q | (Raul,Ortega,Barcells)
X1233254Q | (Laura,Silva,Rubio)
(2 rows)

credit_cards_sl=# select cif, fiscalAddress from Company where id = (select id from CreditCard where associatedAccount = 51235);
cif | fiscaladdress
-----+-----
S0794677B | (Fleming,344,Barcelona,8014,Espanistan)
(1 row)

credit_cards_sl=# select id, billing from Client where id = (select client_id from CreditCard where (cardType).creditLimit = '1000');
id | billing
-----+-----
1 | 1200
(1 row)

credit_cards_sl=# █
```

Update / Delete

```
credit_cards_sl=# update Person set age = 24 where (fullName).name = 'Raul';
UPDATE 1
credit_cards_sl=# update Company set registerDate = now() + interval '5 YEAR' where cif = 'S0754677B';
UPDATE 1
credit_cards_sl=# update CreditCard set cardType = ('3000', 1, 'Premium') where associatedAccount = '51232';
UPDATE 1
credit_cards_sl=# delete from CreditCard where expiryDate < now() + interval '5 YEAR';
DELETE 3
credit_cards_sl=# delete from CreditCard where client_id = (select id from client where billing = 1200);
DELETE 0
credit_cards_sl=# delete from client where billing = 1200;
DELETE 1
credit_cards_sl=# █
```